Fall 2001 Midterm

1. This midterm is due on *Thursday, November 1, 2001* at the beginning of class. Late midterms will *not* be accepted. There are six (6) problems and a total of 100 points.
2. You are to work **alone** on this midterm. You may **not** talk to anyone about questions on this midterm, except for the 6.857 TAs and Prof. Rivest. Collaboration or copying are **not** allowed and will not be tolerated.
3. This exam is open book. You may use any printed or Web resource, provided that you credit the sources properly.
4. Type up your solutions, use separate sheets per problem (subparts of a problem can be on the same page), and put your name and problem number on each page. You will receive 1 point for each problem in which you follow these directions correctly.
5. Each problem has a specified a page limit. Pages beyond the specified limit will be ignored. Do not feel obliged to use the entire allotment; many problems have concise answers.

Problem Q-1. File Systems and Hashing [19 pts]

In this problem we examine a simple read-only file system which has a single directory mapping filenames to database keys[1]. There are no inodes or access control.

Filename	Database key
foo	0x29085324
bar	0x50932152
⋮	⋮

Figure 1: The directory maps filenames to database keys.

Figure 1 shows how a directory might appear. An auxiliary database provided by a publisher uses a hash table (with chaining) to map database keys to file content. To read the file "foo" in the above example, the file system first maps the filename to the database key 0x29085324. The system then asks the database for the value associated with this key. A publisher creates this database and chooses the database keys.

Ben Bitdiddle has modified this file system to provide for integrity protection of file contents. In his file system, a database key is instead the cryptographic hash of the value pointed to. If the file "foo" contains the text "March20", then the database key is hash ("March20"). If one trusts that

[1] By a "key" we mean a key/value pair as in an associative array or hash table

the directory itself has integrity, then a file system can not only locate file data, but also verify the integrity of the file data by recomputing and checking the hash. Furthermore, such a file system saves storage by identifying files that match exactly. If several files have the same content, then the hashes will also match. As a result, the database must store only one key/value pair for all the matching files.

Ben wanted to use SHA-1 as the hash function, but Ben's management required that database keys have minimal length without endangering the security of file integrity. Help Ben select an appropriate hash function output size that balances the database key size with the strength of integrity protection. How might you parameterize the length of the hash output with the strength of integrity protection and the sizes of file system structures? Limit your essay to one page. You can assume there exists a CR hash function for any given output size.

Problem Q-2. Numbing Theory [20 pts]

Ben Bitdiddle has proposed the following encryption scheme for his client, Perry Noyd. Perry has a public key consisting of a large prime p, a generator g modulo p, and a value g^a modulo p; where the secret key a is a randomly chosen value modulo $p-1$.

Ben proposes that Perry's friend Amy encrypt a message $m = m_1 m_2 \dots m_k$ for Perry (where each m_i is a bit) as follows. For each bit m_i, Amy first picks a value b at random modulo $p-1$. She then sends Perry the pair (g^b, g^{ab+m_i}) of values modulo p. Ben says to Perry that this scheme is secure, based on the usual Diffie-Hellman assumption, thus justifying its inefficiency.

1. Ben has suggested to Perry that a should be chosen so that $\gcd(a, p-1) = 1$. Ben claims that this ensures that $g^a \bmod p$ is also a generator. Argue that Ben's claim is correct. Limit your answer to this subproblem to one page.
2. Explain how Perry can decrypt Amy's message. Limit your answer to this subproblem to one page.
3. Explain why Ben's proposed encryption is not very secure; it leaks information about m to an eavesdropper. Limit your answer to this subproblem to one page. (Hint: consider quadratic residuosity)

Problem Q-3. Count Transitivania [20 pts]

In the country of Transitivania, each citizen is required to vote and each citizen is considered eligible for the office of President. Voting is interpreted "transitively", so that if A votes for B, and B votes for C, then it is as if A voted for C directly (and so on, if C has voted for someone else; the chains get followed to the end). A citizen who wishes to be President votes for himself, stopping a chain. If a nontrivial "cycle" develops (e.g., A votes for B, who votes for C, who votes for B) then all the relevant votes are discarded.

1. Sketch an approach towards implementing the Transitivanian voting system that maintains voter privacy as well as you can. Limit your answer to this subproblem to one page. (You may answer however you like, but you might consider as a hint the homomorphic voting scheme described in class.) You need not implement "candidate privacy" (keeping secret who has voted for themselves).

2. What do you think of "transitive voting"? What are the advantages and disadvantages of transitive voting? Limit your discussion of this subproblem to one page.

Problem Q-4. In the know [20 pts]

In many password-based login systems, the server maintains the hash of a password, $h(p)$, and accepts a login if the user is able to produce a preimage, p, for this hash value.

In one of the zero-knowledge protocols that we have studied, a Prover is able to prove to a Verifier that she knows the discrete log, a, for some public value $y = g^a \bmod p$.

Describe the differences between a password, an interactive zero-knowledge proof of knowledge, and a non-interactive zero-knowledge proof of knowledge. Be sure to distinguish between the first and third items. Limit your discussion to one page.

Problem Q-5. Block Cipher [20 pts]

Ben Bitdiddle has designed a simple and fast block cipher algorithm. A diagram of his cipher is presented below. The cipher encrypts blocks of n bits and requires a key of size $3n$ bits. The key is split into three n-bit subkeys, k_1, k_2, k_3. The input block is first XORed with k_1, then added to k_2 and multiplied by k_3. The addition and multiplication operations are performed $(\bmod 2^n)$. It is important to note that k_3 must be odd in order to be able to decrypt properly. In order to guarantee this, the low order bit of k_3 is first ORed with 1.

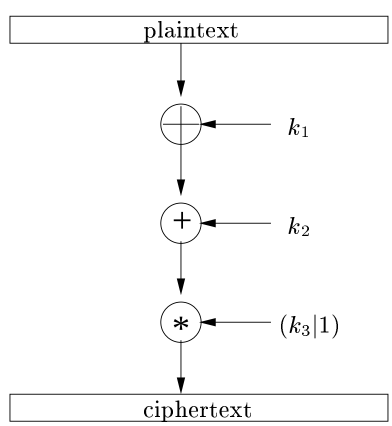

Figure 2: Ben's block cipher

Decryption works by performing the operations in essentially reverse order. In the first step, multiply the ciphertext block by $(k_3|1)^{-1}$ (that is, find the inverse of $(k_3|1) \bmod 2^n$ and multiply by this value), and then subtract k_2. Finally, XOR the resulting value with k_1 to get the plaintext.

Is this block cipher secure against a chosen-message attack? That is, if you are allowed to ask Ben for encryptions of arbitrary messages, are you able to mount an attack that recovers the secret key bits? If so, present the attack. If not, justify. Limit your response to one page.

Problem Q-6. Academic honesty [1 pt]

Write a couple sentences to testify that you have not collaborated with anyone on this midterm and that you have cited all your sources. Affix your pretty signature to this statement.